Analyzing the dynamics of discrete deterministic systems

Luis M Torres, Annegret K Wagler

Analyzing the dynamics of discrete deterministic systems

Luis M. Torres
ModeMat, Escuela Politécnica Nacional
Quito, Ecuador
Email: luis.torres@epn.edu.ec

Annegret K. Wagler
CNRS and LIMOS, Université Blaise Pascal
Clermont-Ferrand, France
Email: wagler@isima.fr

***Abstract*—This work is based on an extension of the Petri net framework. Our model relies on the definition of a priority relation between conflicting transitions, which is encoded in a compact manner by orienting the edges of a transition conflict graph. The benefit is that this allows the use of a successor function for the study of dynamic processes from a global point of view, independent from a particular initial state and the (complete) construction of the reachability graph. We address the problem of gaining the information that allows to provide an appropriate priority relation governing the dynamic behavior of the studied system and discuss some further implications and generalizations of the studied approach.**

I. Introduction

To model complex dynamic systems, Petri nets constitute a well-established framework that offers a variety of analysis techniques. Hereby, the structure of the systems is described by means of a network, while dynamic processes are usually represented in terms of state changes. Several fundamental issues concerning the dynamics of such systems can be addressed within this setting, either via theoretical analysis or computer simulation. Accordingly, Petri nets have a broad application range, including the design of asynchronous hardware circuits [1], the analysis of production and workflow systems [2], the analysis and control of batch processes [3], the design of distributed algorithms for networks of agents [4], and the modeling and simulation of biological networks [5], [6], [7], to cite only some prominent examples.

More formally, a *network* $G = (P, T, A, w)$ reflects the involved components (like network elements, technical components, biological entities etc.) by places $p \in P$ and their interactions (like transformations, causal dependencies, chemical reactions etc.) by transitions $t \in T$, linked by weighted directed arcs.

Some places $B \subseteq P$ may have bounded capacities, which are given by a positive integral vector $u \in \mathbb{Z}_+^B$. Each place $p \in P$ can be marked with an integral number x_p of tokens (at most u_p if $p \in B$). Any such marking defines a state of the system that can be represented as an integral nonnegative vector $x \in \mathbb{Z}_+^P$. The *potential state space* of a capacitated network (G, u) is the set of all theoretically possible states

$$\mathcal{X} := \{x \in \mathbb{Z}_+^P : x_p \leq u_p,\ \forall p \in B\},$$

if no further conditions restrict $\mathcal{X}$. Its size is at least exponential in $|P|$ and is finite if $B = P$ holds.

Dynamic processes are described as sequences $x^0, x^1, \ldots, x^k$ of consecutive system states[1], where state x^{i+1} is obtained from x^i by switching a transition $t \in T$. Thereby, t consumes w_{pt} tokens from each place p with $(p,t) \in A$ and produces w_{tp} new tokens on each place p with $(t,p) \in A$. A transition $t \in T$ is enabled at a state $x \in \mathcal{X}$ if switching t yields a valid successor state in $\mathcal{X}$. We denote by $T(x)$ the set of enabled transitions at state x and, accordingly, by $X(t)$ the set of states at which transition t is enabled.

Usually, the dynamic behavior is the result of several conflicting as well as concurrent ongoing dynamic processes. A *Petri net* is a pair (G, x^0) consisting of a network G together with an initial state x^0, and its state space $\mathcal{X}(x^0)$ is understood as the set of all further system states which can be reached from x^0 by switching or firing sequences of transitions, see e.g. [8] for more information. Describing the dynamics of a system might be done by model animation, i.e., by simulating the flow of tokens inside the network as transitions are switched. Given a network G and an initial state x^0, some central problems are:

- *Reachability*: Determine whether the system may reach one of a set of target states after a finite sequence of transition switches.
- *Boundedness*: Determine if there are sequences of transition switches that lead to the accumulation of an unlimited number of tokens at some place.
- *Liveness*: Determine whether no sequence of transition switches can put the system into a state where some transition is permanently disabled.

These problems are in general hard from a computational complexity point of view. For instance, *reachability* was proven to require exponential space [9] and decidability of this problem could only be established some years later [10], [11].

a) Partial versus global point of view on a system: Dynamic processes can be encoded as directed paths in a state digraph $\mathcal{G}$ where nodes represent states and there is a directed arc between two states x, x' if x' can be obtained from x by switching a single transition. It is common practice to use the term *reachability* or *marking graph* to refer to the subgraph $\mathcal{G}(x^0)$ of $\mathcal{G}$ induced by the set $\mathcal{X}(x^0)$ of nodes corresponding to those states that can be reached from the initial state x^0 of the network.

[1] We use superindices to reference different states and subindices to specify places. Thus, x_p^i is the number of tokens assigned to place p at state x^i.

The marking graph $\mathcal{G}(x^0)$ allows only a partial view on the studied system which is, e.g., suitable for a technical system performing exactly one process. However, already a failure of one or several components can change the initial state, and a more global view is required to coherently model both normal and malfunction of the system, and to detect the faulty element(s) by an according failure analysis.

Similarly, the study of biological systems by performing experiments can be seen as putting the system in an initial state and observing the evolution of the system in terms of sequences of state changes. Due to the intrinsic complexity of biological systems, the dynamic behavior of such systems can rarely be understood by performing a single experiment so that, in general, several experiments starting from different initial states are required.

To study complex dynamic systems and processes therein from a more global point of view (independent from a particular initial state), we therefore suggest to consider the state digraph $\mathcal{G}$ on the potential state space $\mathcal{X}$ of the system, i.e., on the set of all theoretically possible states.

b) Non-deterministic versus deterministic systems: While studying dynamic processes, a particular situation occurs when so-called dynamic conflicts are present at states, in which two transitions are enabled, but switching one disables the other. In this case, model animation does not allow definite conclusions about any system properties, as mentioned in [12]. The reason is that the occurrence of dynamic conflicts is understood as alternative (branching) system behavior, where a decision between these alternatives is taken non-deterministically.

However, there are examples of systems that show a deterministic behavior despite the existence of dynamic conflicts. We call a dynamic system *deterministic* if any state $x \in \mathcal{X}$ has a unique successor state $\mathrm{succ}(x)$. For technical systems, a deterministic behavior is often crucial in order to guarantee the reliability of the performed processes.

In addition, also some biological systems are deterministic, as stimulating them in a certain way triggers always the same response (see e.g. the light-induced sporulation of *Physarum polycephalum plasmodia* or the phototaxis of halobacterial cells described in [13], [14], [15]). Petri nets, as originally defined, can be used to model such systems only in some trivial cases.

c) A compact encoding for deterministic systems: To overcome the above mentioned difficulties, we propose to model deterministic systems from a global point of view (independent from a particular initial state). For that, a successor function $\mathsf{succ} : \mathcal{X} \to \mathcal{X}$ can be defined which returns $\mathrm{succ}(x)$ for every potential state $x \in \mathcal{X}$. Note that an explicit encoding of succ, for instance pointwise, is exponential in the size of the network G.

If the change from a state $x \in \mathcal{X}$ to its successor state $\mathrm{succ}(x)$ can always be explained by the switch of a *single* transition, the dynamics of a deterministic system can be alternatively expressed with the help of a *transition selection function* $\mathsf{trans} : \mathcal{X} \to T$. This function assigns to every state $x \in \mathcal{X}$ a unique transition $\mathsf{trans}(x) \in T(x)$ that must be switched in order to reach $\mathrm{succ}(x)$.

In [16], it was proposed to use priorities between the transitions[2] of the network as additional activation rules to determine which transition from $T(x)$ has to be selected as $\mathsf{trans}(x)$ in order to reach $\mathrm{succ}(x)$. In [17], the following compact scheme for encoding trans based on such priorities was introduced. The *transition conflict graph* of (G, u) is an undirected graph $\mathbb{K} = (T, \mathbb{E})$ having as nodes the transitions from G, where two transitions t, t' are joined by an edge if and only if there exists at least one state where both are enabled.

A directed graph $\mathbb{D}$ obtained by orienting the edges of $\mathbb{K}$ is a *valid orientation* for (G, u) if, for every state $x \in \mathcal{X}$ with $T(x) \neq \emptyset$, the node subset $T(x)$ has a unique sink, and this sink coincides with $\mathsf{trans}(x)$. Observe that the size of $\mathbb{D}$ is $O(|T|^2)$, which is polynomial in the size of G, see Section II.

A valid orientation completely encodes the dynamic behavior of a deterministic system: given a state $x \in \mathcal{X}$, determine the highest-priority transition $\mathsf{trans}(x)$ by searching for the unique sink in the subgraph of $\mathbb{D}$ induced by $T(x)$; switching $\mathsf{trans}(x)$ yields the successor $\mathrm{succ}(x)$.

Computing successors is a key operation within simulation algorithms to study the dynamic behavior of a system and to address questions like reachability, boundedness or liveness. To the best of our knowledge, currently no algorithm for these problems is known, which does not rely on storing explicit descriptions of the state digraph or some equivalent structure to compute $\mathrm{succ}(x)$ at every state x, which limits the size of the networks that can be considered, due to the high memory requirements. Having, however, a valid orientation allows to determine $\mathrm{succ}(x)$ for each state $x \in \mathcal{X}$.

d) Experiment design to obtain such models: To benefit from the above mentioned properties, a valid orientation of a transition conflict graph needs to be obtained. This shall be done by observing dynamic processes in the underlying deterministic system: based on the knowledge of $\mathrm{succ}(x)$ at *some* states $x \in \mathcal{X}$, we aim at finding a valid orientation that encodes the global dynamic behavior of the system since $\mathrm{succ}(x)$ can be determined for *all* states $x \in \mathcal{X}$.

Querying $\mathrm{succ}(x)$ for a state x is typically done by performing experiments with the studied system. Since this can be expensive and time-consuming, it is worth to design the experiments in such a way that as few of them suffice to gain the required information. In this paper, we address a problem in this context posed in [18]: given a network G and an *oracle* for returning $\mathrm{succ}(x)$ for any state $x \in \mathcal{X}$, we aim at determining $\mathbb{D}$ by calling the oracle as few times as possible. Thus, we shall design experiments (in terms of oracle calls) to deduce the global dynamic behavior of the system by providing enough transition priorities, encoded in $\mathbb{D}$. In Section III, we formally state the problem, present some related concepts from [18] and propose an algorithm to solve the problem.

We close with some concluding remarks and a discussion on potential generalizations of our approach.

[2]These priority relations shall reflect the relative reaction rates of the (chemical) reactions represented by the transitions of the network with the idea that faster reactions have higher priorities and are taken. On the model side, priorities can be seen as a discrete extreme case of firing rates of transitions defined by probability distributions, where exactly one transition in $T(x)$ (namely, the highest-priority transition $\mathsf{trans}(x)$) has probability 1, and all other transitions in $T(x)$ have probability 0.

II. Encoding valid orientations

As observed in the previous section, one key issue for modeling the dynamic behavior of a deterministic system is the specification of a mechanism for the (unambiguous) resolution of dynamic conflicts between enabled transitions.

Recall that, given a capacitated network $G = (P, T, A, w)$ with $u \in \mathbb{Z}_+^B$, its *transition conflict graph* is an undirected graph $\mathbb{K} = (T, \mathbb{E})$ having as nodes the transitions from G, where two transitions t, t' are joined by an edge if and only if there exists at least one state where both are enabled, i.e.,

$$tt' \in \mathbb{E} \quad \Leftrightarrow \quad X(t) \cap X(t') \neq \emptyset.$$

As the involved sets $X(t)$ are boxes, $\mathbb{K}$ can be constructed in $O(|P|\ |T|^2)$ time using a straightforward algorithm to check for box intersections $X(t) \cap X(t')$. A more efficient way of its computation is proposed in [17].

It follows from the definition of $\mathbb{K}$ that, for every state $x \in \mathcal{X}$, the set $T(x)$ of enabled transitions induces a clique in $\mathbb{K}$, i.e., a set of mutually adjacent nodes.

Remark 1: The converse is not necessarily true, as it is shown in [17]. However, as the sets $X(t)$ are boxes, it is possible to prove that at least the inclusion-wise maximal cliques in $\mathbb{K}$ are associated with states of the system.

Example 2: Consider the capacitated network $(G, \mathbb{1})$, with $G = (P, T, A, \mathbb{1})$, from Figure 1(a). It has $\mathcal{X} = \{0, 1\}^4$ as set of potential system states, where for each $x^i \in \mathcal{X}$, with $0 \leq i < 16$, it holds that $i = x_1^i 2^0 + x_2^i 2^1 + x_3^i 2^2 + x_4^i 2^3$.

The sets $X(t)$ for all transitions in T are as follows:

$$\begin{array}{lcl} X(t_1) & = & \left\{x^1, x^3, x^5, x^7\right\}, \\ X(t_2) & = & \left\{x^2, x^3, x^6, x^7\right\}, \\ X(t_3) & = & \left\{x^5, x^{13}\right\}, \\ X(t_4) & = & \left\{x^4, x^5, x^6, x^7\right\}. \end{array}$$

The resulting transition conflict graph $\mathbb{K}$ is shown in Figure 1(b). Note that $(G, \mathbb{1})$ has four branching states and the following cliques of $\mathbb{K}$ are associated with them:

$$\begin{array}{l} T(x^3) = \{t_1, t_2\}, \\ T(x^5) = \{t_1, t_3, t_4\}, \\ T(x^6) = \{t_2, t_4\}, \\ T(x^7) = \{t_1, t_2, t_4\}. \end{array}$$

$T(x^5)$ and $T(x^7)$ are the inclusion-wise maximal cliques of $\mathbb{K}$. Moreover, there are five cliques of size two (edges), from which only two (t_1t_2 and t_3t_4) are associated with states.

$\mathbb{K}$ is a plausible candidate to embed priorities between transitions. This motivates the following definition. A directed graph $\mathbb{D} = (T, \mathbb{A})$ obtained by orienting the edges of $\mathbb{K}$ is *valid for the system* (G, u) if, for every state $x \in \mathcal{X}$ with $T(x) \neq \emptyset$, the subgraph induced by the nodes from $T(x)$ has a unique sink, and this sink coincides with $\mathsf{trans}(x)$.

The existence of a valid orientation of $\mathbb{K}$ implicitly imposes a further requirement on the nature of a dynamic system: if two states $x, x^* \in \mathcal{X}$ have the same set $T(x) = T(x^*)$ of enabled transitions, then both induce the same subgraph of $\mathbb{D}$ and, therefore, $\mathsf{trans}(x) = \mathsf{trans}(x^*)$ must hold.

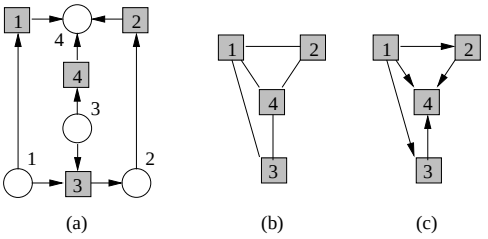

Fig. 1. (a) A capacitated network of a dynamic system (places are represented by circles, transitions by squares, all place capacities and arc weights are assumed to be equal to one), (b) the corresponding transition conflict graph, (c) a valid orientation.

Moreover, if a transition t is enabled at two states $x, x' \in \mathcal{X}$ and t is the highest-priority transition at x, then either t is also the highest-priority transition at x' or $\mathsf{trans}(x') \notin T(x)$. The next result from [17] shows that this condition is also sufficient for the existence of a valid orientation.

Theorem 3: A system (G, u) has a valid orientation if and only if the following *consistency condition* holds: For any pair of states x and x', if $\mathsf{trans}(x) \in T(x) \cap T(x')$ then *either* $\mathsf{trans}(x') = \mathsf{trans}(x)$ *or* $\mathsf{trans}(x') \in T(x') \setminus T(x)$.

Recognizing whether an orientation $\mathbb{D}$ of $\mathbb{K}$ is valid is hard in general, since it requires a test for every state $x \in \mathcal{X}$, whether the corresponding clique $T(x)$ in $\mathbb{D}$ has a unique sink.

Note that a clique cannot have more than one sink, and that every clique without a sink contains a directed cycle. Hence, if $\mathbb{D}$ is acyclic then every clique of the graph has a unique sink, and we immediately obtain:

Observation 4: Every acyclic orientation of $\mathbb{K}$ is valid.

Example 5: For the capacitated network $(G, \mathbb{1})$ from Figure 1(a), the priority relation $t_1 < t_2, t_1 < t_3, t_2 < t_4, t_3 < t_4$ induces an acyclic orientation $\mathbb{D}$, depicted in Figure 1(c), of the transition conflict graph $\mathbb{K}$ from Figure 1(b). $\mathbb{D}$ is indeed a valid orientation: each of the cliques in $\mathbb{D}$ associated with (branching) states has a unique sink:

$$\begin{array}{l} \mathsf{trans}(x^3) = t_2 \text{ in } T(x^3), \\ \mathsf{trans}(x^5) = t_4 \text{ in } T(x^5), \\ \mathsf{trans}(x^6) = t_4 \text{ in } T(x^6), \\ \mathsf{trans}(x^7) = t_4 \text{ in } T(x^7), \end{array}$$

and the consistency condition from Theorem 3 is satisfied:

- $\mathsf{trans}(x^3) = t_2$ belongs to the intersection of $T(x^3) \cap T(x^6)$ and $T(x^3) \cap T(x^7)$, but $\mathsf{trans}(x^j) = t_4$ belongs to $T(x^j) \setminus T(x^3)$ for $j = 6, 7$;
- t_4 is the common highest-priority transition in $T(x^5), T(x^6), T(x^7)$.

III. Finding valid orientations for a network

A relevant problem when modeling dynamic systems with our approach consists in inferring a valid orientation for the transition conflict graph from observations made on dynamic processes in the underlying deterministic system. Based on the knowledge of the values of $\mathrm{succ}(x)$ at *some* states $x \in \mathcal{X}$, we aim at finding a valid orientation that allows to predict $\mathrm{succ}(x)$ for *all* states $x \in \mathcal{X}$.

Given as input the capacitated network (G, u) and an *oracle* for returning the value of $\mathsf{trans}(x)$ at any state $x \in \mathcal{X}$, we shall determine $\mathbb{D}$ by calling the oracle as few times as possible since in practice, a call to the oracle stands for the execution of an (expensive and time-consuming) experiment. A set $\mathcal{X}' \subseteq \mathcal{X}$ of states is a *valid test set* if the information about the corresponding highest-priority transitions $\{\mathsf{trans}(x) : x \in \mathcal{X}'\}$ is sufficient for inferring the direction of all arcs in $\mathbb{D}$. We are interested in the following problem, formulated in [18].

Problem 6 (Minimum Valid Test Set Problem (MVTP)): Given a deterministic system together with an oracle for returning highest-priority transitions, find a valid test set of minimum cardinality.

In the following, we discuss and extend an approach from [18] to solve the MVTP iteratively, by alternating oracle calls and analyzing the gained information.

For that, suppose that we are in the situation of having already a *partial orientation* of a (mixed) graph $\mathbb{D}' = (T, \mathbb{A}', \mathbb{E}')$, obtained by fixing the direction for *some* edges of $\mathbb{K}$ (e.g. as result of previous calls to the oracle) with $\mathbb{A}'$ being the set of the corresponding oriented arcs, and $\mathbb{E}'$ the set of the remaining unoriented edges. A partial orientation is *extendible* if it is possible to choose directions for all unoriented edges in such a way that a valid orientation is obtained. Given an extendible partial orientation $\mathbb{D}' = (T, \mathbb{A}', \mathbb{E}')$, a yet unoriented edge $tt' \in \mathbb{E}'$ is said to be *inferable as* (t, t') if the partial orientation $(T, \mathbb{A}' \cup \{(t', t)\}, \mathbb{E}' \setminus \{tt'\})$ is not extendible and *dominated* if for all states x with $t, t' \in T(x)$, $\mathsf{trans}(x)$ is already known to differ from t and t'. Observe that the direction of dominated edges is irrelevant in our model, as these never provide information for computing highest-priority transitions. However, as we shall see later, the direction of dominated edges may be important in order to determine which of the other edges are inferable and which are not. We call $\mathbb{D}'$ to be *sufficient* for inferring a valid orientation $\mathbb{D} = (T, \mathbb{A})$ if all edges in $\mathbb{E}'$ are inferable as the corresponding arcs in $\mathbb{A}$.

Example 7: Figure 2 illustrates these concepts. The partial orientation depicted in

(a) is not extendible, as choosing any direction for edge t_1t_4 produces one inclusion-wise maximal clique without sink;

(b) is extendible but not sufficient: $t_1 < t_4$ is inferable, but then t_1t_3 is not inferable;

(c) is sufficient: $t_1 < t_4$ is inferable and implies $t_3 < t_4$;

(d) is sufficient since $t_1 < t_4$ is inferable.

This also shows that the optimal solution of MVTP may depend on the underlying valid (complete) orientation $\mathbb{D}$, and not on the transition conflict graph alone.

Testing whether a partial orientation is extendible or not is computationally hard, for the same reason for which recognizing a valid orientation is hard in general. In the following, we focus on the particular case where the valid (complete) orientation $\mathbb{D}$ is known *a priori* to be acyclic. This is equivalent to assuming that there exists a partial orientation reflecting the priority relations among all transitions of the network, an assumption which is plausible in many practical applications.

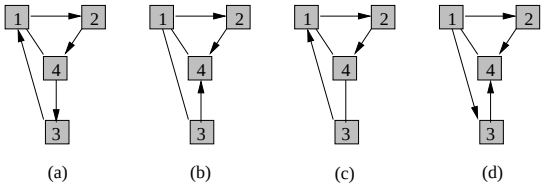

Fig. 2. Different types of partial orientations: (a) non-extendible; (b) extendible, but non-sufficient; (c) and (d) sufficient.

Note that any acyclic partial orientation $\mathbb{D}'$ can be extended to a (complete) acyclic orientation $\mathbb{D}$. Thus:

Observation 8: Every acyclic partial orientation is extendible.

Furthermore, a partial orientation $\mathbb{D}'$ is sufficient if, for every non-dominated edge $tt' \in \mathbb{E}'$, the digraph $(T, \mathbb{A}')$ contains either a directed path from t to t', or a directed path from t' to t. Conversely, given a (complete) orientation $\mathbb{D} = (T, \mathbb{A})$, an arc $a = (t, t') \in \mathbb{A}$ such that the digraph $(T, \mathbb{A} \setminus \{a\})$ does not contain a directed path from t to t' is either dominated or otherwise *essential*.

Observe that the direction of an essential arc a cannot be inferred in any partial orientation extendible to $\mathbb{D}$, whereas the orientation of a dominated arc can be arbitrarily chosen.

Any sufficient partial orientation must therefore contain the set $\mathbb{A}^*$ of all essential arcs from $\mathbb{D}$. Furthermore, for any state $x \in \mathcal{X}$, the information on $\mathrm{succ}(x)$ can be used to orient at most one essential arc among the set

$$A(x) = \{(t, \mathsf{trans}(x)) : t \in T(x),\, t \neq \mathsf{trans}(x)\}$$

of arcs whose directions are revealed by testing the system at x. As a consequence, the following result is obtained in [18] for systems satisfying the *clique property*: each clique of the transition conflict graph is associated with a state of the system.

Theorem 9: If a system satisfies the clique property, then

- the essential arcs of $\mathbb{D}$ form a sufficient partial orientation;
- a valid test set $\mathcal{X}' \subseteq \mathcal{X}$ is optimal for MVTP if and only if $A(x) \cap \mathbb{A}^* \neq \emptyset$ holds for every $x \in \mathcal{X}'$.

Here, we extend the results from [18] to the case that $\mathbb{D}$ is known *a priori* to be acyclic, but the clique property is not assumed to hold. Recall that this is reasonable to expect in practical instances.

Theorem 10: If the complete orientation $\mathbb{D}$ is acyclic, then

- the essential and the dominated arcs of $\mathbb{D}$ form a sufficient partial orientation;
- a valid test set $\mathcal{X}' \subseteq \mathcal{X}$ is optimal for MVTP if $A(x) \cap \mathbb{A}^* \neq \emptyset$ holds for every $x \in \mathcal{X}'$.

Algorithm 1 shows an approach to determine a valid test set in this case. It computes as first step the set $\mathcal{Q}$ of all inclusion-wise maximal cliques in $\mathbb{K}$ (to ensure that each clique in $\mathcal{Q}$ is associated to a state), then it alternates oracle calls and the analysis of the gained information.

Input: (G, u, trans) {deterministic system with oracle for computing trans}
$\mathbb{K}$ {transition conflict graph}
Output: $\mathbb{D}$ {valid acyclic orientation}
initialize $\mathcal{Q}$ as the set of all inclusion-wise maximal cliques in $\mathbb{K}$
while $\mathcal{Q} \neq \emptyset$ **do**
  retrieve a clique Q from $\mathcal{Q}$
  while Q contains more than one node and there is $x \in \mathcal{X}$ with $T(x) = Q$ **do**
    call the oracle and determine $t^* := \mathsf{trans}(x)$
    orient the arcs $\{(t, t^*) : t \in T(x), t \neq t^*\}$
    deduce orientations for inferable edges
    remove t^* from Q
  end while
  if Q contains more than one node **then**
    compute $\mathcal{Q}' := \{T(x) \subset Q : x \in \mathcal{X}\}$
    remove from $\mathcal{Q}'$ cliques that are not inclusion-wise maximal
    $\mathcal{Q} := \mathcal{Q} \cup \mathcal{Q}'$
  end if
end while
$\mathbb{D}' := (T, \mathbb{A}', \mathbb{E}')$ is the partial orientation obtained so far
while $\mathbb{E}' \neq \emptyset$ **do**
  deduce an orientation for all yet unoriented inferable edges
  choose an arbitrary orientation for a yet unoriented dominated edge
end while

Algorithm 1: Inferring an acyclic valid orientation.

Theorem 11: If the orientation of $\mathbb{D}$ is known to be acyclic, then Algorithm 1 determines a valid test set for inferring $\mathbb{D}$.

Example 12: In order to determine a valid test set $\mathcal{X}' \subseteq \mathcal{X}$ to infer a valid orientation of the edges in $\mathbb{K}$ from Figure 1(b), we recall from Example 2 that the following cliques of $\mathbb{K}$ are associated with states:

$$T(x^3) = \{t_1, t_2\},$$
$$T(x^5) = \{t_1, t_3, t_4\},$$
$$T(x^6) = \{t_2, t_4\},$$
$$T(x^7) = \{t_1, t_2, t_4\}.$$

The set of all inclusion-wise maximal cliques in $\mathbb{E}$ is $\mathcal{Q} = \{T(x^5), T(x^7)\}$. Applying Algorithm 1 to $\mathbb{K}$ and $\mathcal{Q}$ yields:

For $Q = Q(x^5)$, we call the oracle in x^5 and receive $\mathrm{succ}(x^5) = x^9$. We conclude $\mathsf{trans}(x^5) = t_4$ and, thus, $t_4 > t_3$ and $t_4 > t_1$ follows. After removing t_4, the dominated edge t_1t_3 remains. We compute $\mathcal{Q}' = \emptyset$, so no new cliques are added to $\mathcal{Q}$.

For $Q = Q(x^7)$, we call the oracle in x^7 and receive $\mathrm{succ}(x^7) = x^{11}$. We conclude $\mathsf{trans}(x^7) = t_4$ and, thus, $t_4 > t_2$ and $t_4 > t_1$ follows. After removing t_4, we obtain $Q = \{t_1, t_2\}$. This clique is associated with x^3, so we call the oracle in x^3 and receive $\mathrm{succ}(x^3) = x^9$. We conclude $\mathsf{trans}(x^3) = t_2$ and, thus, $t_2 > t_1$. After removing t_2, the remaining subclique has only one node.

The resulting partial orientation $\mathbb{D}'$ is depicted in Figure 3(a). It is acyclic and extendible to the valid orientation $\mathbb{D}_1$ in Figure 3(b) or $\mathbb{D}_2$ in Figure 3(c) by assigning any direction to the dominated edge t_1t_3.

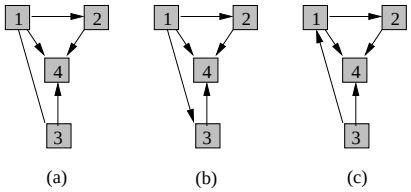

Fig. 3. (a) The partial orientation $\mathbb{D}'$ obtained by Algorithm 1; (b) and (c) valid orientations obtained by assigning any direction to the dominated edge t_1t_3.

It is left to discuss the quality of the test set obtained by Algorithm 1. For that, we firstly note the following.

Lemma 13: Given a capacitated network (G, u) with acyclic orientation $\mathbb{D}$, then the number of essential arcs of $\mathbb{D}$ (resp. branching states of (G, u)) is a lower (resp. an upper) bound for the cardinality of a minimum valid test set.

Example 14: Consider the two acyclic valid orientations $\mathbb{D}_1$ in Figure 3(b) and $\mathbb{D}_2$ in Figure 3(c). In both cases, t_1t_3 is a dominated edge. The essential arcs are

- t_1t_2, t_2t_4, t_3t_4 in $\mathbb{D}_1$;
- t_1t_2, t_2t_4 in $\mathbb{D}_2$.

These essential arcs form together with the dominated edge t_1t_3 the two sufficient partial orientations depicted in Figure 2(d) and Figure 2(c), respectively. For $\mathbb{D}_1$, the number of essential arcs equals the cardinality of the test set $\mathcal{X}'$ obtained by Algorithm 1 (thereby verifying its optimality). $\mathbb{D}_2$ shows that both bounds do not need to be tight (we have $|\mathbb{A}_2^*| = 2$, but $|\mathcal{X}'| = 3$ and four branching states x^3, x^5, x^6, x^7).

This illustrates the difficulties for certifying optimality of a computed test set. However, we conjecture the following:

Conjecture 15: Given a capacitated network (G, u),

- any acyclic valid orientation $\mathbb{D}$ contains the partial acyclic orientation $\mathbb{D}'$, computed in the first part of Algorithm 1;
- there is one acyclic valid orientation $\mathbb{D}^*$ whose set of essential arcs equals $\mathbb{D}'$;
- if $\mathbb{D}^*$ is the valid orientation encoding the dynamic behavior of (G, u), then the test set used by Algorithm 1 is optimal.

IV. Discussion

In this paper, we have examined a new approach for encoding the dynamic behavior of certain deterministic discrete systems that relies on extending the familiar framework of Petri nets. Our encoding consists in a realization of the successor-function succ as a valid orientation of the edges of the transition conflict graph $\mathbb{K}$. This encoding is compact in the sense that the amount of space required for its storage is polynomial in the size of the network. Therefore, it is well-suited for being integrated into simulation algorithms like [19] to study the dynamics of large complex deterministic systems, and to address issues such as reachability, boundedness, existence of deadlocks, and liveness, among others.

To benefit from the above mentioned properties, a valid orientation of a transition conflict graph needs to be obtained. This shall be done by observing dynamic processes in the underlying deterministic system: based on the knowledge of $\mathrm{succ}(x)$ at *some* states $x \in \mathcal{X}$, a valid test set, we aim at finding a valid orientation that encodes the global dynamic behavior of the system since $\mathrm{succ}(x)$ can be determined for *all* states $x \in \mathcal{X}$. In this paper, we address the MVTP: given a network G and an *oracle* for returning $\mathrm{succ}(x)$ for any state $x \in \mathcal{X}$, we aim at determining $\mathbb{D}$ by calling the oracle as few times as possible. Thus, we shall design experiments (in terms of oracle calls) to deduce the global dynamic behavior of the system by providing enough priorities, encoded in $\mathbb{D}$. In Section III, we presented some related concepts from [18] and proposed an algorithm to solve the problem for systems where it is known that valid orientations are acyclic.

A possible interesting extension of our model concerns the study of dynamic systems where the concurrent switch of various transitions can occur. Throughout this paper we have assumed that a change from a state $x \in \mathcal{X}$ to its successor state $\mathrm{succ}(x)$ can always be explained by the switch of a *single* transition $\mathsf{trans}(x)$. However, there are deterministic systems where this does not hold, but a transition selection function

$$\mathsf{trans} : \mathcal{X} \to 2^T$$

is required that assigns to each state x a *subset* of transitions to be switched in order to reach $\mathrm{succ}(x)$, see [18] for examples.

One way of dealing with these systems could be through the inclusion of pseudo-nodes in the transition conflict graph, to account for the simultaneous switching of various transitions. The advantage would be that all results from the previous sections directly carry over to the extended setting. The disadvantage of such a modification is, however, the dramatic increase in the size of the transition conflict graph, as the number of required pseudo-nodes may grow exponentially with respect to $|T|$ (in the worst-case upto $2^{|T|}$ according to the number of subsets of T).

An alternative approach consists in working on a slightly different transition conflict graph, where an edge between two transitions means that they are in dynamic conflict (i.e. when switching one transition disables the other). In this case, $T(x)$ does not longer induce a clique in $\mathbb{K}$. Yet we can now define an orientation to be valid if for any state $x \in \mathcal{X}$ the set $T(x)$ of enabled transitions induces a subgraph containing *at least one* sink. These sinks reveal an *anti-chain* of transitions with maximal priorities, exactly the subset $\mathsf{trans}(x) \in 2^T$, which have to be switched concurrently. Theorem 3 can be generalized to this new setting in a straightforward manner, characterizing which deterministic systems admit valid orientations. The advantage of this scheme is that it is still compact with respect to the size of the network G. However, as the sets $T(x)$ do not induce cliques in the conflict graph, many of the combinatorial properties pointed out in Section II do not hold any more. Moreover, since all transitions that *may* switch concurrently are *required* to do so, it is again possible to construct examples of deterministic systems whose behavior cannot be modeled in this way. Hence, further research is necessary to propose a compact encoding of the successor function for general deterministic systems.

For a large number of such systems, however, the here presented results allow us already the use of such an oracle for the study of dynamic processes from a global point of view, independent from a particular initial state and the (complete) construction of the reachability graph.

REFERENCES

[1] A. Yakovlev, A. Koelmans, A. Semenov, and D. Kinniment, "Modelling, analysis and synthesis of asynchronous control circuits using Petri nets." *Integr., VLSI J.*, vol. 21, no. 3, pp. 143–170, 1996.

[2] N. R. Adam, V. Atluri, and W.-K. Huang, "Modeling and analysis of workflows using Petri nets," *J. Intell. Inf. Syst.*, vol. 10, no. 2, pp. 131–158, 1998.

[3] T. Gu and P. A. Bahri, "A survey of Petri net applications in batch processes," *Comput. Ind.*, vol. 47, no. 1, pp. 99–111, 2002.

[4] W. Reisig, *Elements of distributed algorithms: modeling and analysis with Petri nets*. New York, NY, USA: Springer-Verlag New York, Inc., 1998.

[5] S. Hardy and P. N. Robillard, "Modeling and simulation of molecular biology systems using Petri nets: modeling goals of various approaches," *Journal of Bioinformatics and Computational Biology*, vol. 2, no. 4, pp. 619–637, Dec 2004.

[6] I. Koch and M. Heiner, "Petri nets," in *Analysis of Biological Networks*, B. H. Junker and F. Schreiber, Eds. Wiley Book Series in Bioinformatics, 2008, pp. 139–180.

[7] C. Chaouiya, E. Remy, and D. Thieffry, "Petri net modelling of biological regulatory networks," *J. of Discrete Algorithms*, vol. 6, no. 2, pp. 165–177, 2008.

[8] W. Reisig, *Petri nets: an introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.

[9] R. Lipton, "The reachability problem requires exponential space," Yale University, Computer Science Dept., Research Report 62, 1976.

[10] E. W. Mayr, "An algorithm for the general Petri net reachability problem," in *Proceedings of the 13th Ann. ACM Symposium on Theory of Computing*. ACM Press, 1981, pp. 238–246.

[11] ——, "An algorithm for the general Petri net reachability problem," *SIAM J. Comput.*, vol. 13, no. 3, pp. 441–460, 1984.

[12] M. Heiner, D. Gilbert, and R. Donaldson, "Petri nets for systems and synthetic biology," in *Formal methods for computational systems biology*. Springer, 2008, pp. 215–264.

[13] W. Marwan and C. Starostzik, "The sequence of regulatory events in the sporulation control network of physarum polycephalum analysed by time-resolved somatic complementation of mutants," *Protist*, vol. 153, pp. 391–400, 2002.

[14] W. Marwan, "Theory of time-resolved somatic complementation and its use for the analysis of the sporulation control network of physarum polycephalum," *Genetics*, vol. 164, pp. 105–115, 2003.

[15] W. Marwan, A. Sujatha, and C. Starostzik, "Reconstructing the regulatory network controlling commitment and sporulation in physarum polycephalum based on hierarchical Petri net modeling and simulation," *Journal of Theoretical Biology*, vol. 236, pp. 349–365, 2005.

[16] W. Marwan, A. Wagler, and R. Weismantel, "A mathematical approach to solve the network reconstruction problem," *Math. Methods of Operations Research*, vol. 67, pp. 117–132, 2008.

[17] L. M. Torres and A. Wagler, "Encoding the dynamics of deterministic systems," *Math. Methods of Operations Research*, vol. 73, no. 3, pp. 281–300, 2011.

[18] ——, "Model reconstruction for discrete deterministic systems," *Electronic Notes of Discrete Mathematics*, vol. 36, pp. 175–182, 2010.

[19] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick, "Snoopy a unifying petri net tool," in *Application and Theory of Petri Nets*, ser. Lecture Notes in Computer Science, S. Haddad and L. Pomello, Eds. Springer Berlin Heidelberg, 2012, vol. 7347, pp. 398–407.